DOCUMENT RESUME

ED 343 417 FL 020 144

AUTHOR Higgins, John
TITLE Power to the Pupils: Computer Software in Language Teaching.
PUB DATE 88
NOTE 13p.; In: Materials for Language Learning and Teaching. Anthology Series 22; see FL 020 138.
PUB TYPE Speeches/Conference Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Classroom Techniques; *Computer Assisted Instruction; *Computer Software; English (Second Language); Foreign Countries; Instructional Effectiveness; *Instructional Materials; *Material Development; Second Language Instruction; *Second Language Learning

ABSTRACT

A discussion of computer-assisted second language learning focuses on the kinds of computer software currently available and discusses briefly how they can be used in language instruction. Four types of software are distinguished, including: (1) that in which the computer controls, to a great extent, the nature and order of events (e.g., drills, quizzes, exercises, tests, and programmed learning); (2) insertion exercises, in which some piece of information is missing and must be supplied, an innovative approach; (3) tools for editing, correction, etc., which should be made available to students as well as teachers and administrators; and (4) materials containing content but imposing no structured methodology. It is concluded that what computers can do, in general, is supply optimal challenge to students, offering opportunities for students to discover things that even the teacher doesn't know. Additional examples of the four software types are appended, with the addresses of sources supplied. (MSE)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED343417

Power to the Pupils: Computer Software in Language Teaching

John Higgins

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Chen, J.

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

X This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

FL020441

BEST COPY AVAILABLE

POWER TO THE PUPILS; COMPUTER SOFTWARE IN LANGUAGE TEACHING

John Higgins

I am going to talk about teachers and learners, about computers and programmes, hardware and software. I am going to demonstrate any machinery. For the next 45 minutes you will be looking not at hardware or software but at liveware.

My brief for this talk was to say a little about the methodological principles behind CALL, to describe available and recommended software for language learners and to talk about trends, in other words to indulge in prophecy. The central part of the assignment produces a list, and since a list makes a boring lecture I have put that into a handout which I shall talk about only briefly, leaving me more time to speculate and to talk about how learners use software rather than talking about the software. Machines themselves are of no importance, or should not be; they are the paper on which the software is written and, just as you only think about paper when it becomes so torn that you can't read it, so you only think about machines when they are so difficult to use and unreliable that you cannot use them. Unfortunately that happens all too often, but they are getting better all the time, better in the sense of being unobtrusive and unimportant.

I'll begin with an anecdote. Twelve years ago I was conducting an oral examination in English for a group of pupils in Bangkok. Most of them had reached the standard you would expect; they could use English, but their vocabulary was limited, they made a lot of grammar errors, and their pronunciation was heavily influenced by Thai. There was one student, however, who was different. He understood everything I said; his accent was almost perfect, and he hardly made any grammar errors. What made him different? I looked at his record form, but could find nothing to explain it. He had not lived overseas and did not seem to have attended any particularly well-known schools. "Who were your teachers?" I asked him, thinking he might have had a native speaker in his school class. "My teachers?" he said, "my teachers were Richard Burton, Laurence Olivier, Vivien Leigh and Gregory Peck." He was a cinema buff, and watched movies with the kind of empathy that led him to imitate the behaviour of the screen heroes and heroines, including their language behaviour.

Now a naive reaction to a success story like this might be to say: school language classes are useless; let us instead spend all the hours for English on the timetable at the cinema. We could run a controlled experiment, perhaps, with

one group getting ordinary textbook work and the other group seeing classic films. Then we could publish learned articles showing that the boys in the cinema group performed 2.131% better than the control group and that this is significant at the p=.99 level, while the girls showed only 1.7% gains and this is significant at the p=.9 level. (I have something of a prejudice against people who fill their papers with obscure mathematics).

A more sensible reaction would be to notice that what distinguished this pupil was his enthusiasm, his love not so much of English as of some users of English. English for him had become a hobby rather than just a subject. Once you have harnessed that kind of enthusiasm and alertness to a subject, then there is no further problem in getting the subject learned. The quality of the teacher and the materials hardly matters; old-fashioned materials serve just as well as the most modern ones.

There are, I think, many teachers who would be uncomfortable to have that student in their class. It is usually unpleasant to have learners who do not succeed in learning what you teach them, but it may be even more disturbing to have learners who learn what you do not teach them. Yet there is bound to be a mismatch between teaching and learning (using learning for the time being in a broad sense which includes acquisition). If there were not, then education would be a continuous process of attrition, since you cannot pass on the totality of your knowledge. What we know about language as competent (not necessarily native) speakers could not possibly be stated explicitly in a finite time.

Let me try a little experiment to illustrate this. I want you to imagine you are doing a grammar exercise, a straightforward sentence transformation task. If you have a pen and a piece of paper handy, try writing down your response. When I tell you what to do, try doing it without too much thought; just write down the first response that occurs to you without agonising over whether it is the one I want.

Here is the task: make this sentence singular

They write books.

Now what did you write? Some of you wrote

She is writing a book. (or He is writing a book.)

doing exactly what I asked you to. Some of you may have written

She writes a book.

but then worried about it, perhaps crossed it out. The point is that I did not tell you to change the verb aspect, and yet that is what you would automatically do if you were thinking about meaning rather than form.

If we consider the four sentences

She writes books.
She is writing a book.
She writes a book.
She is writing books.

all four of them are grammatical, but the first two are natural, while the third and fourth require rather elaborate contextual justification. I could dream up contexts for them, for instance for number three:

She keeps getting divorced, yet she still manages to look cheerful. How does she do it?

Oh, every time she ditches a husband, she writes a book.

For number four:

Sally used to go to the bridge club regularly, but she has given that up. She's writing books instead.

If you want a linguistic explanation, you could say that there is a semantic feature +/- specific which applies to both verb phrase and noun phrase. A noun phrase with "a" can be either, but it defaults to +specific, ie we expect the speaker to have a particular book in mind unless the context makes that unlikely. On the other hand the plural, "books", defaults to -specific. Similarly the continuous verb aspect, "is writing" defaults to writing in mind, whereas the simple present, "writes" defaults to + specific; we can expect the speaker to have a particular act of writing in mind, whereas the simple present, "writes" defaults to -specific. Now it doesn't really matter whether you made that analysis or even understood it. What matters is that you instinctively would have chosen the right form in any meaningful context. How did you do it? Nobody ever taught it to you; it is not in the textbooks as a grammar rule. If textbooks came down to this kind of detail, they would weigh tons and cost fortunes. And that brings us to a characteristic of textbooks that has not been stressed very much yet this week, the fact that they are necessarily meagre. The quantity of teaching they contain is insufficuent to do the job unaided. As managers of learning we have to make sure that we open windows on a wider world of language outside the textbook.

The outside world, and even some people within the language teaching profession, seem to have what I call the Barber's Shop image of education. You want your hair cut? OK, go and sit down on that chair, turn your head the way the barber tells you, and wait till he or she has finished cutting. You want to learn English? OK, sit on that bench. Open your mouth when the teacher tells you to, and wait until he or she has finished teaching.

I am exaggerating, of course. Nobody seriously imagines that learning is quite as passive a process as getting a haircut. However, this kind of thinking accounts for the spectacular failure of certain kinds of CALL. One claim that used to be made often was that computers would let students make up work that they had missed through illness. (The same claims had been made for programmed learning and language laboratories.) Another claim was that one could learn at home. The strongest claim was that made by Tom Stonier that in future home would be the place one went to learn, and school would be the place for socialising. Yet any child who has tried to make up more than a few hours of missed work, or any parent who has tried to buy their offspring a real headstart by getting a computer and some disks, will know that it doesn't work. It is easy to blame the machines and the software. But do you think that a compulsory (note the compulsory) series of visits to the cinema to see old Gregory Peck movies would work any better? One cannot simply give a child a disk and say: "Away you go and learn your irregular verbs." Learning involves involvement. Parents who succeed in improving their children's education are those who learn alongside them.

So the software we like is software which involves teachers, learners, and parents in the same enterprise. We distrust the kind of software that tries to do the whole job, that tries to foresee every possible student error and supply an appropriate feedback message for each. We believe that the software is only one element of a learning environment that includes many other components, such as the dormant knowledge of the individual, what you can work out or remember if you are not under threat of public humiliation, such as the pooled knowledge of a group who collaborate on a task, such as the use of a teacher as a consultant, "Come and tell us why this did not work", and plain old-fashioned reference books. Learners often succeed in making software change its nature by using it in new ways; drills become games, toys become tools, and tools become toys. Long may this continue. In particular drill activities which are timed are sometimes turned spontaneously into races, and anything which involves scoring becomes a competition, either a competition against others or one against oneself, a constant effort to improve on one's best scores. Classroom teachers can also harness the competitive instinct among learners, but the language laboratory, with its pre-recorded fixed response, could never do this.

So what kinds of software are there? I have put down a rough-and ready classification into four main types, though I doubt whether every programme could

be made to fit the scheme comfortably. There is one major category missing, the artificial intelligence group in which the machine functions as expert system or as tutee, learning from inputs and showing, often in its failures, something about the complexity of natural language. The reason for omitting this was that I do not know of any significant published programs in this area which can be used as they stand by learners, apart from the ubiquitous ELIZA. Watch this space.

Category A is "DO WHAT I TELL YOU", the machine controlling to a great extent the nature and order of events. This includes drills, exercises, quizzes and tests, and at the most organised end of this spectrum programmed learning. The mechanics are familiar. The machine gives you a task, say a sentence to complete or a question to answer. You can answer by typing out a word or phrase, or by selecting an answer in a multiple choice format, say by pressing a number key. The machine tells you if you are right or wrong and then invites you to try again if you were wrong. There may be some feedback about why the answer was wrong as soon as you have found the right answer. These programmes, just like books, can be good or bad, error-free or errorful, relevant of irrelevant.

However, one very interesting point has emerged from recent experience, which is embodied into some of the best software, particularly Wida Software's TESTMASTER. The conventional assumption used to be that you commented on wrong answers but, as soon as a student had found a right answer you said "Well done" and moved on to the next item. What Chris Jones, the author of TESTMASTER, has pointed out is that students have not necessarily lost interest in an item just becuase they got it right. Perhaps they would like to know what is wrong with a different answer which they were thinking of giving. this is a point which class teachers should take note of, not just programmers.

My second category, GUESS WHAT WAS THERE, is perhaps the clearest innovation that CALL has supplied in its short life. Tim Johns invented the form with his TEXTBAG program, in which all the words of a text were masked out and you had to point to single words and buy them, buying the minimum number of words that you needed in order to answer a comprehension question displayed at the end of the text. I followed this up with STORYBOARD, in which the words of a masked out text had to be guessed, and each occurrence of a successfully guessed word was supplied wherever it occurred, in technical terms a type-replacement activity rather than a token replacement activity. STORYBOARD probably has the distinction of being the most widely imitated CALL programme ever; there must be more than a dozen versions current. I could talk for a couple of hours about the programme and about what is done with it, but - I shall not. I do, however, urge any one who has never seen a version running to try it out; you could find it changes your whole perception of teacher/learner roles. In fact all the programmes in this category and in the next one belong to

Dr Prâbhu's description of "meta-materials", namely structured activities with no content of their own; the content can be supplied by teachers and, whisper it, learners.

My third category, "CAN I HELP YOU?", is purely the computer as tool. If we as writers or administrators find them useful, why should we cut our students off from the benefits? Let them explore the language with their aid. Let them have the satisfaction of producing attractively presented work that they can display with pride, edited and corrected (by the teacher and by each other) with none of the tedium of making a fair copy by hand. For all that we may value hard work, it is generally the case that when we make work easier, we get more of it done. Whatever the pocket calculator may have done to our mental arithmetic skills, it has certainly led to more calculations being performed. It has ceased to be tedious to compare the cost of fabrics sold by the metre for dollars with those sold by the yard for pounds.

The fourth category, which I called "HOW DO I GET OUT OF THIS?", consists of activities which Dr Prabhu might call semi-materials, ie materials which have content but impose no structured methodology. Many of them were not created for language learners at all. Even those that were, such as our own INVENTION program, often lead to the comment, "But that's a maths program, not an English program". The answer to that, of course, is that we live in a world full of mathematics, the shape of a building, the speed of a car, the cost of a glass of beer. But we talk about these concepts in language, and we are not obliged to limit such talk to the maths lesson. (At this conference Ballagh and Moore have already given us a wealth of ideas for getting language out of mathematics and vice versa). We should always be trying to open a window on a world in which language can be acquired, a world where language is used in a concentrated way to receive and transmit meanings.

So finally to predictions. Where are we going from here? Perhaps you are expecting me to talk about laser disks and interactive video, about CD-ROMs and a world where the whole of the Library of Congress can be consulted from one's bedroom by pushing a few keys, about tele-conferencing, the global village of instant communication. These things will come, will be exciting, will take a little time to absorb, and will eventually be taken for granted. Meanwhile what of the learner and the teacher? Learners can use these things just as easily as anybody else. Machines supply power. However, we will need to learn how to use it. Although we will be able to get answers to a infinity of questions, we need to train ourselves to ask more questions. When we have the training we will have power to learn from a much wider part of our environment than a single teacher can provide.

The learning instinct is strong in all normal circumstance; we are constantly engaged in an effort to make sense of our environment, and in making predictions. There are two cases in which this learning instinct will be suppressed.

8 BEST COPY AVAILABLE

One is when the environment is too complex to be explained; that is what happens to me when I hear conversation in Chinese. The other is where the environment is so impoverished and predictable that it contains nothing which requires explanation, no problems to be solved and no uncertainties to be predicted. This might occur if I had to carry out an automatic drill on sentences whose meaning doesn't matter. Both cases, too much challenge and too little, lead to the same outcome, boredom or the switching off of the learning instinct.

What computers are good at, because of the rapid and slavish way they respond, is supplying optimal challenge, something a teacher with forty students can hardly ever do. Optimal challege is individually determined and can vary from minute to minute; at times I will want to be stretched, at others to receive all the help that is available. With the computer as a slave, learners and small groups of learners can readily find optimal challenges. This, more than any other reason, accounts for the so-called addictiveness of computers, the fact that even children with short attention spans can spend long periods with them.

The outcome is not likely to be that everyone will uniformly learn more and improve their performance. Since computers supply opportunities, not fixed paths to follow, improvement will be sporadic and unpredictable; we are likely to encounter more individuals like my star pupil in Thailand without necessarily seeing an overall jump in standards. The pioneers of CALL saw themselves as handicappers in a horse race, bringing all the horses to the finishing line abreast. What we predict instead is that computers will act like adrenalin, sending some of the horses racing far ahead of the field. In the process teachers are going to find their authority, in particular their authority as 'knowers' of their subject, will be undermined, since some of the learners will seize opportunities to discover things the teacher does not know. Teachers will have to find new roles, as advisers, as managers, even as fellow learners discovering new insights into language by using the same facilities as their students. For some teachers it will be an uncomfortable experience. Good luck!

APPENDIX: TYPES OF SOFTWARE

A. "DO WHAT I TELL YOU"

Quizzes - These are useful reading practice even when they are not specifically about language. there are quiz programmes in the public domain, including many which allow you to enter your own questions. Often the best use of them is to ask learners to write new questions and use them to challenge other members of the class.

Tests - eg Longman's SCREENTEST. These are much more popular with learners than one expects them to be. One needs to be careful, since a great many programmes in circulation, especially those in the public domain, are badly written and contain blatant errors or are confusingly worded.

Exercises - one can find ready-made exercises, but there are many programmes which allow you to build your own, eg Wida Software's GAPMASTER, TESTMASTER, and MATCHMASTER, and RDA's MARK-UP (for punctuation). Building good exercises is hard work; at least these programmes allow teachers to learn from their own mistakes and improve the exercises gradually.

Programmed learning - very difficult to write, though programmes to help you write branching programmes, such as Strange Software's INFO, can be bought very cheaply. The main problem is that of supplying relevant feedback to learners who have made mistakes. Basically this is something machines are not very good at; neither, for that matter, are books.

B. "GUESS WHAT WAS THERE"

Text reconstruction exercises are superficially like other kinds of exercise, except that the nature of the challenge is different; you have to reconstruct by trial and error a particular piece of prose rather than find a correct form of sentence.

There are basically four ways of mutilating a text and challenging learners to reconstruct it: your can delete things from it; or your can re-order it. You can carry out these operations at the level of the letter, the word, or the sentence, and you can do it partially or totally.

STORYBOARD, RHUBARB, QUARTEXT and CLOZEMASTER are all deletion programs, the first three being total deletion, ie everything in the text has gone. DOUBLE-UP, GOON, SEQUITUR and TEXTPLAY are examples of re-ordering programs.

Almost all programs in this category allow you to enter your own pieces of text. This makes them what Dr Prabhu described as 'meta-materials'. Teachers can select and enter text which is relevant to course work, but do not forget that

they also allow learners to enter their own texts to challenge each other; this can provide a strong motivation for composition.

A slightly different type of deletion exercise is PINPOINT, which shows a tiny fragment of text and invites you to guess the title; if you do not want to guess, you can ask to see more of the text, and so on until you can guess confidently. The object of this is to persuade readers that they can understand something even if they cant't understand everything, and to help them develop a better risk-taking strategy in their ordinary reading.

In general the computer is a poor medium for reading long texts or for developing any of the skills of extensive reading. Most of the so-called reading skills programs, in which you have to read a text on screen and then answer comprehension questions, would be better done on paper. However, there may be diagnostic applications, and my own HOPALONG program attempts to help learners discover something about their own reading style and about the kinds of text they can cope with.

C. "CAN I HELP YOU?"

The computer's natural role is that of a slave, obeying orders and carrying out jobs on demand. The obvious language job is word-processing or, in other words, displaying fair copy of the master's second thoughts.

Cheap public domain and shareware word-processors are easy to find. One of the most interesting is MINDREADER, which says it is designed for bad typists; it tries to guess the word you are typing and, as soon as you have typed two or three letters, displays a list of possible words so that you can finish the word by typing a number. If you carry on typing instead of choosing a word, it will remember the word, so that it learns your style and soon starts offering the words you yourself use frequently.

All word-processors have a "search-and-replace" function, which can be used to create practice material from any piece of text which has been typed in. You can replace all the articles with "____" and then print out the text so that the learners can write them back in. For a more difficult task, use a piece of first-person narrative and replace all the first- person pronouns with "____". Now ask learners to re- write the text as third-person narrative, putting appropriate pronouns or names into the spaces and making any other necessary changes.

More expensive word-processors include outliners, spelling-checkers, thesauruses, and grammar- or style- checkers. These are of very little use as writing aids, but they are fascinating as ways of exploring the language, as programmes to play with. Feed in a nonsense word and watch what alternatives the spelling-checker offers you. Look at the set of synonyms offered by the thesaurus and

then look for synonyms of the synonyms. Discuss which words will fit the context, and what is wrong with the others. Type in some Shakespeare and watch the style-checker offer its naive comments.

The most powerful aid of all for letting learners play with language is a concordancer, ie a program which finds particular words in a text and prints out the line on which the owrd occurs. You can use the FIND command in MSDOS for this purpose. WORD COUNT EXTRA from Strange Software produces a list of all the words used in a text, sorted either alphabetically or by their frequency. True concordance programs produce key-Work-in-Context (KWIC) displays, where you see the key word in the centre of the screen with half a line of context before and after, making it much easier to notice the way words come together, eg which preposition occurs after "believe" or the fact that the word "same" is almost always preceded by "the".

D. "HOW DO WE GET OUT OF THIS?"

Simulations can take many forms, but can be classified as "real-time" and "move-based". In the former an event is in progress which you can control but which requires some action from you; you can fail by being too slow. Examples are flight simulators, all the arcade action games, and sports simulations. They do not usually generate much language while they are in progress, but they can be talked about or written about afterwards.

Move-based simulations present you with a situation and ask you to make a decision. The computer calculates the outcome of your decision and then asks for the next decision. The action freezes while you decide what to do. There are many business and trading simulations of this form, such as CUP's FAST FOOD. Experience of using these with classes shows that the spoken language of the discussion is rather impoverished; people do not say more than they need to, and at worst it comes down to barking numbers at each other. You can improve things greatly by having the discussion take place away from the computer screen, and appointing a "runner" who enters the group's decisions and reports back to them the computer's reponse.

Adventures are a special kind of simulation, often based on well-known stories, in which you take on the role of a character, are given an objective, and proceed by giving commands to the machine or by selecting choices. Most advantures are too difficult for most EFL learners (or teachers) to solve, but CUP's LONDON ADVENTURE is designed for EFL learners, and there may be others which can be adapted.

Logic problems can also generate spoken language, and can be directed at specific points of usage. RDA's INVENTION, for instance, provides unlimited meaningful examples of sentences using comparatives in a problem setting. We

are currently engaged in writing more programmes of this type to activity in which "play" can be given a strongly grammatical focus is fortune- telling and horoscopes; fortune-telling programmes are easy to find, and they give plenty of opportunities for using future tenses and if-clauses.

Sources of software mentioned here (all for MSDOS machines):

Research Design Associates, P O Box 848, Stony Brook, New York, USA

Wida Software, 2 Nicholas Garden, London W5 UK

Cambridge University Press, The Pitt Building, Cambridge, UK

Strange Software Ltd, 40 Bowling Green Lane, London EC1R ONE

Public Domain and Shareware libraries: the addresses are given in computer magazines.